CPSC 221: Written Assignment 2

Last Updated: October 28, 2016(changed title) Due: 21.00 Monday November 7, 2016.

Submission Instructions

Handin your solutions using `handin`. You can write your solutions by hand and scan the pages or take pictures of them with your phone; or use a word processing package to typeset your solutions and produce a .pdf file.

To handin: Copy the files that contain your solutions to the directory `~/cs221/assign2` in your home directory on an undergraduate machine. (You may have to create this directory using `mkdir ~/cs221/assign2`.) Then run `handin cs221 assign2` from your home directory.

We encourage you to work in pairs. **Be sure to include the names and ugrad login IDs of both partners on all solution pages, but only one partner should handin the assignment.**

Late submissions are accepted subject to the following penalty: You lose $100 \times (2^{\lfloor m/5 \rfloor})/64$ percent of your mark, where m is the number of minutes late your assignment is. For example, if you hand in 10 minutes late, you'll lose $100 \times (2^2)/64 = 6.25\%$ of the mark, but if you hand in 25 minutes late, you'll lose 50% of the mark. You cannot hand in more than 30 minutes late.

Acknowledge all collaborators or sources of assistance (besides the course staff, handouts, and required textbooks) on the first page of your assignment by name. If you quote from or derive work from any source, you *must* acknowledge that source where it is used as is usual for citations. We don't need a formal citations list, although that's not too hard to produce with LaTeX.

Questions

1. (5 points) Convert the unordered array

8	11	9	6	7	2	4	5	10	1	3

into a minimum heap using the *heapify* algorithm shown in class. Show the array and the associated binary tree after each (non-recursive) call to *swapDown*.

2. (5 points) Describe a **tail recursive** (not iterative) algorithm in pseudocode that counts the number of unmatched closing parentheses `')'` in a string. For example, `"(hello (world)) ()) (oops!))"` has two unmatched closing parentheses, while `") totally unmatched ("` has one unmatched closing parenthesis. You may use a helper function.

3. Lightning struck the offices of BitSort Inc. scrambling the code for their patented product: a piece of C++ code that sorts arrays of 0's and 1's **by swapping**. You've been hired by BitSort Inc. to reconstruct this code and prove to the company lawyer that it works **using a loop invariant**. Here's what remains of the code:

```
// Sort an array A of 0's and 1's so the 0's come before the 1's
// using swapping.  The size of the array is n.
int bitsort(int *A, int n) {
  int i=0;
  int j=n-1;
  while( ????? ) {
    if( ????? ) {
      i++;
    } else if( ????? ) {
      ?????;
```

```
      } else {
        swap( ?????, A[j] );
      }
    }
    return j;
  }
```

(a) (5 points) Replace each ????? with C++ code. Do not add any additional lines of code and do not replace ????? with multiple statements (in other words, don't use ";").

(b) (5 points) State and prove a loop invariant for the while-loop that you can use to establish the correctness of your code.

(c) (2 points) Prove that the correctness of your code follows from the loop invariant and the termination condition of the loop.

(d) (2 bonus[1] points) Under what conditions does your code return the number of 0's in the input array?

4. Another use of invariants is in analyzing strategies for playing games. Suppose a necklace with $n \geq 1$ links is part of a huge treasure. Two players play a game to see who gets the treasure. Players alternate turns. In each turn, a player can remove one or two links from the necklace. The winner of the treasure is the player who removes the last link. Let Alice be the player with the first turn and Bob the player with the second turn.

   (a) (5 points) Prove by induction on n that Alice can always win if n is not divisible by three. What invariant (property of the remaining links) does Alice preserve in order to win?

   (b) (5 bonus[1] points) Now suppose that each player must remove one link or two *originally adjacent* links in their turn. Prove that Bob can always win if $n \geq 3$. What invariant does Bob preserve in order to win?

5. Suppose we use the following hash function to hash strings of 8-bit characters, using $m = 2^{16} - 1$:

$$h(s_0 s_1 \ldots s_k) = (s_0 + s_1 \cdot 256 + s_2 \cdot 256^2 + \cdots + s_k \cdot 256^k) \mod m$$

   (a) (5 points) Show that there are 257 three-character strings that all hash to the same slot. You don't need to specify these strings, just argue that they exist.

   (b) (5 points) When you try this hash function, you might notice that some permutations of a string of distinct characters hash to the same slot in the hash table (but some do not). For instance, $h(\text{ABCD}) = h(\text{CDAB}) \neq h(\text{BACD})$. Describe which permutations of a string of distinct characters hash to the same slot.

[1] Your answer must be completely correct to earn any bonus points. You cannot get partial bonus points.